*Andrés Felipe Borrero (15203033) y Juan Camilo Tobar Núñez (15203036).*

**DISEÑO DE CASOS DE PRUEBA MUNDO**

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 1: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que al inicializar los símbolos estos sean diferentes de null y su colección sea mayor a cero en tamaño.

**Clase:** AgenteDeBolsa

**Método:** inicializarSimbolos (): void

**Escenario:** Escenario1, que inicializa únicamente el Agente de bolsa.

**Valores de entrada:** Archivo de texto plano.

**Resultado:** Símbolos inicializados correctamente según la información del archivo de texto plano.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 2: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verificar que los corredores se carguen correctamente a través de la importación de un archivo serializado.

**Clase:** AgenteDeBolsa

**Método:** cargarCorredores (): void

**Escenario:** Escenario2, que crea un nuevo Agente de bolsa y carga los corredores en segunda instancia.

**Valores de entrada:** Archivo serializado.

**Resultado:** Se cargan correctamente los corredores, y se asigna exitosamente el corredor actual como el primer corredor cargado.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 3: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que cuando agrega diferentes corredores, el método no lance ninguna excepción sabiendo que todos tienen diferentes username.

**Clase:** AgenteDeBolsa

**Método:** agregarCorredor (Corredor, Corredor): void

**Escenario:** Escenario3, que inicializa un Agente de bolsa que agrega 4 diferentes corredores sin username repetidos.

**Valores de entrada:** Información de cada corredor y el corredor raíz.

**Resultado:** El agente de bolsa agrega exitosamente los corredores sin necesidad de lanzar excepción de algún tipo.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 4: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que la operación de agregar en el agente de bolsa lance una excepción esperada de tipo Ya Existe Corredor.

**Clase:** AgenteDeBolsa

**Método:** agregarCorredor (Corredor, Corredor): void

**Escenario:** Escenario4, que crea un nuevo Agente de bolsa e intenta agregar dos corredores con igual username.

**Valores de entrada:** Información de cada corredor y el corredor raíz.

**Resultado:** Lanzamiento exitoso de la excepción Ya Existe Corredor Exception debido a las condiciones de uso.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 5: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que cuando se busque un corredor, este tenga las mismas características del creado en el escenario.

**Clase:** AgenteDeBolsa

**Método:** buscarCorredor (String, Corredor): Corredor

**Escenario:** Escenario3, que inicializa un Agente de bolsa con 4 corredores agregados correctamente.

**Valores de entrada:** Username del corredor a buscar y corredor raíz.

**Resultado:** El método toString (), que describe toda la información del encontrado, es coherente con el corredor esperado.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 6: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que cuando se busca un corredor inexistente, no se encuentre ninguno y el resultado sea null.

**Clase:** AgenteDeBolsa

**Método:** buscarCorredor (String, Corredor): Corredor

**Escenario:** Escenario3, que tiene un nuevo Agente de bolsa con 4 corredores agregados exitosamente.

**Valores de entrada:** Username inexistente y corredor raíz.

**Resultado:** Corredor resultado de la operación es igual a null.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 7: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que el método de buscar la posición de un activo en el portafolio del corredor, mediante una búsqueda binaria, retorne la locación correcta y al pedir el activo si sea el indicado.

**Clase:** Corredor

**Método:** getPosicionBinarioActivo (String, int, int): int

**Escenario:** Escenario3, que crea un nuevo corredor y se transan exitosamente 4 activos diferentes para su portafolio.

**Valores de entrada:** Símbolo a buscar en el portafolio, la primera posición y la última.

**Resultado:** Posición correcta del símbolo a buscar.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 8: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que la operación de transar se realice sin ningún tipo de fallas, debido a condiciones óptimas.

**Clase:** Corredor

**Método:** transar (Activo, int, String): void

**Escenario:** Escenario1, que inicializa un nuevo corredor con un activo ya transado exitosamente en su portafolio.

**Valores de entrada:** Activo a transar, cantidad a ser transada y tipo de trade.

**Resultado:** El balance después de la compra es el esperado y la cantidad de lotes del activo comprado también.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 9: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que la operación de transar lance excepción de tipo Cantidad No Disponible, debido a falta de lotes o unidades del activo a vender.

**Clase:** Corredor

**Método:** transar (Activo, int, String): void

**Escenario:** Escenario1, que inicializa un nuevo corredor con un activo, del mismo a vender, ya transado exitosamente en su portafolio.

**Valores de entrada:** Activo a vender, cantidad a transar y tipo de trade (Vender).

**Resultado:** Excepción de tipo Cantidad No Disponible lanzada correctamente por falta de lotes de Plata ref. Común.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 10: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que cuando se transa una cantidad de lotes o unidades de cierto activo y el precio de la operación supera el balance del corredor, se lance una excepción de tipo Dinero Insuficiente.

**Clase:** Corredor

**Método:** transar (Activo, int, String): void

**Escenario:** Escenario2, que tiene un nuevo corredor con 100.000 US de balance.

**Valores de entrada:** Activo a transar, cantidad y tipo de trade (Comprar).

**Resultado:** Excepción lanzada exitosamente debido a fondos insuficientes para la compra de la cantidad de unidades del activo.

**\_\_\_\_\_\_\_\_\_\_\_ Caso # 11: \_\_\_\_\_\_\_\_\_\_\_**

**Objetivo:** Verifica que la operación de ordenamiento del portafolio mediante inserción sea completamente exitosa y organice los activos según su símbolo de forma ascendente.

**Clase:** Corredor

**Método:** ordenarPortafolio (): void

**Escenario:** Escenario3, que tiene 4 diferentes tipos de activos transados de forma correcta.

**Valores de entrada:** Ninguno.

**Resultado:** Los activos son ordenados en el portafolio de forma exitosa de acuerdo al criterio de ascendencia por símbolo.